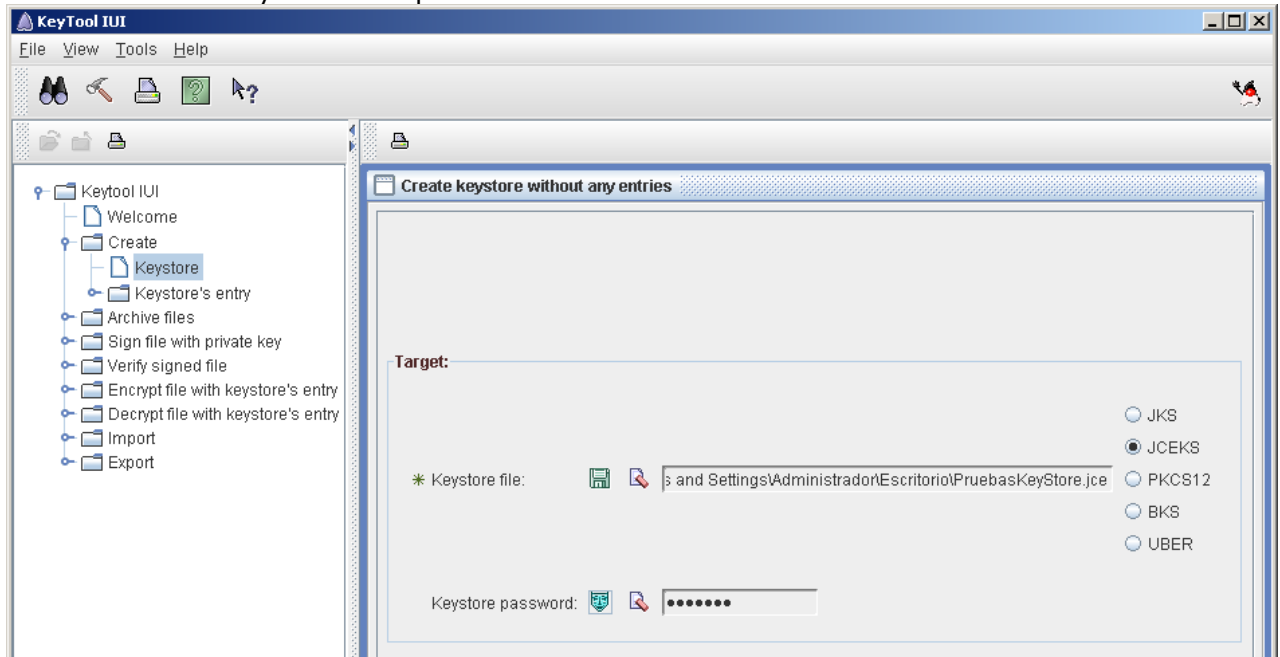


Práctica: Configurar SSL en el servidor JBoss

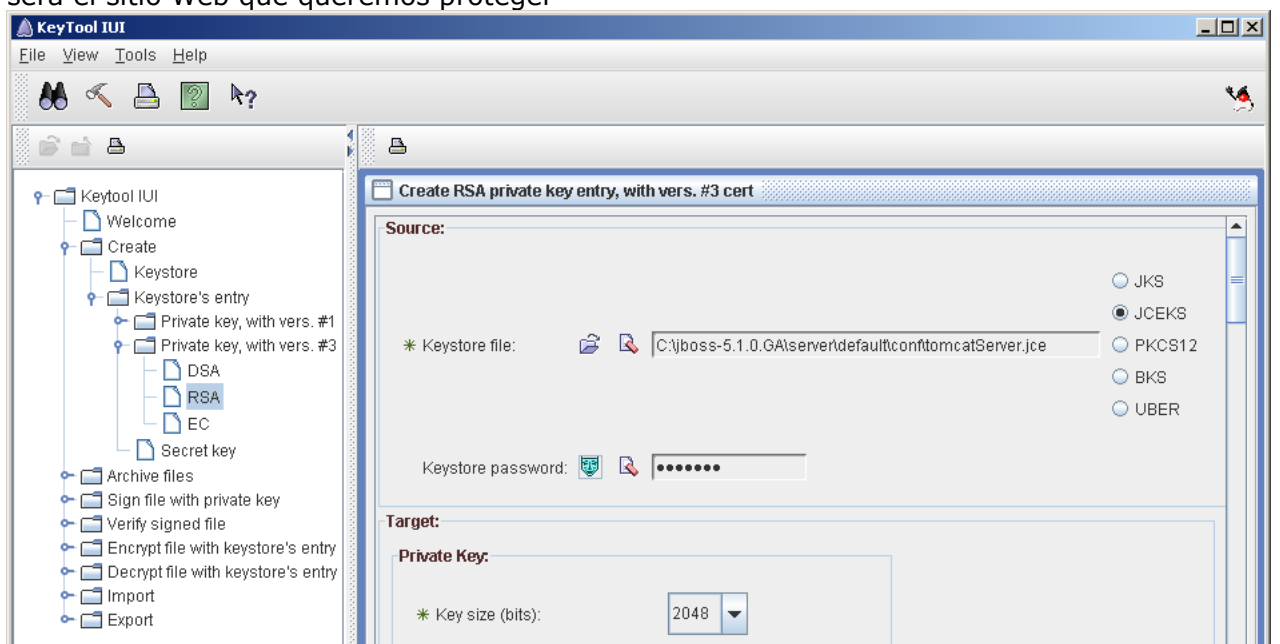
Objetivo: Generar un almacén de claves y generar un certificado para su instalación y configuración en JBoss

Desarrollo:

1. Arrancar la aplicación KeyTool IUI, una interfaz gráfica para keytool
 - 1.1. Crear un nuevo KeyStore de tipo JCEKS



- 1.2. Crear un certificado digital V3 utilizando el keystore previamente definido, el algoritmo utilizado para la firma será SHA1 con RSA, escribir como common name: www.atos.es, que será el sitio Web que queremos proteger



1.3. Introducir un alias y una clave para el certificado creado

Enter new private key entry's alias:


Enter new password:

Confirm new password:

1.4. El certificado aparecerá en el almacén

KeyTool IUI - view JCEKS keystore

Private Key (keypair) & Trusted Certificate Entries:

Alias	Entry	Valid Date ?	Self-Signed ?	Trusted C.A. ?	Key Size	Cert. Type	Cert. Sig. Algo.	Modified Date
serverkey		✓	✓	-	2048 bits	X.509	SHA1withRSA	02-jun-2011

2. Colocar el almacén de certificados en la carpeta conf del servidor JBoss

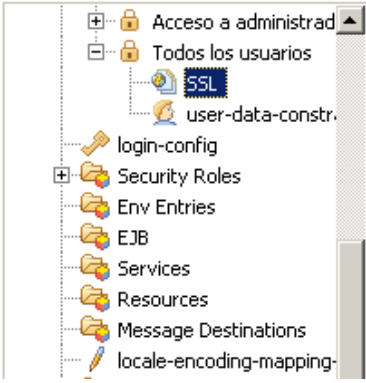
3. Añadimos el conector SSL en el fichero server.xml situado en:

"C:\jboss-5.1.0.GA\server\default\deploy\jbossweb.sar\"

```
<Connector protocol="HTTP/1.1" SSLEnabled="true" port="8443" address="{jboss.bind.address}"
  scheme="https" secure="true" clientAuth="false" keystoreFile="{jboss.server.home.dir}/conf/tomcatServer.jce"
  keystorePass="password" sslProtocol="TLS" keystoreType="JCEKS" />
```

4. Añadimos a una de nuestras aplicaciones Web una restricción para todos los recursos obligando (con una User Data Constraint Confidencial) a ir por SSL.

▼ web



▼ Properties Editor

web resource collection

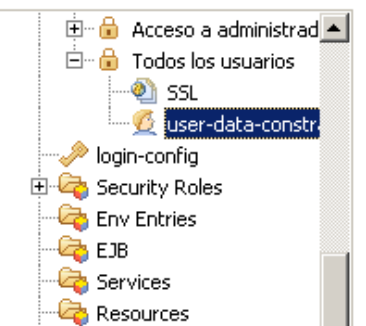
Web-Resource-Name:

URL-Patterns:

Http-Methods:

Description:

▼ web



▼ Properties Editor

user data constraint

Transport-Guarantee:

Description: